Re-routing and resequencing in multistage interconnection networks

Alain Jean-Marie

INRIA

UNITÉ DE RECHERCHE
INRIA-SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France

Tél. (1) 39.63.55.11

Rapports de Recherche

N° 666

RE-ROUTING AND RESEQUENCING IN MULTISTAGE INTERCONNECTION NETWORKS

Alain JEAN-MARIE

Mai 1987

RE-ROUTING AND RESEQUENCING IN MULTISTAGE INTERCONNECTION NETWORKS

RE-ROUTAGE ET RESÉQUENCEMENT DANS LES RÉSEAUX D'INTERCONNEXION MULTI-ÉTAGES

Alain JEAN-MARIE
INRIA, Centre de Sophia Antipolis
Avenue E. Hugues
06560 VALBONNE

RESUME

Nous étudions une variante d'un algorithme de répartition de la charge dans les réseaux de type Omega, dû à Mitra et Cieslak. Le but de cet article est d'analyser le coût de l'algorithme de reséquencement, nécessaire pour remédier aux éventuels dépassements de paquets, conséquence du re-routage aléatoire. Nous obtenons les valeurs analytiques du temps de traversée du réseau et du temps de reséquencement. A travers plusieurs exemples, nous montrons que dans la plupart des cas, le re-routage améilore les performances du système, même en tenant compte des effets du reséquencement.

ABSTRACT

This paper is concerned with load balancing in Omega-Like Networks. A variation of the Mitra-Cieslak random re-routing algorithm is considered. The global aim of the paper is to analyze the cost of the resequencing algorithm that has to be implemented in order to cope with possible message overtaking due to re-routing. Analytical expressions for the statistics of both network response time and resequencing time are obtained. It is shown through several examples that in "most situations", such random re-routing improves the global network performance, even when including the effects of resequencing.

KEYWORDS Omega Network - Load Balancing - Resequencing

0/INTRODUCTION

In multi-processor systems design, as well as in communication networks, the problem of load balancing is of primary interest.

Fully connected networks are rarely a realistic solution to the problem of interconnecting numerous stations, since physical constraints can prevent some nodes to be directly connected. When networks are not fully connected, certain different node-to-node communications need to pass through a same link, which will cause blockings, in the case of a circuit-switching mode, or additional delays in the case of packet-switching mode. In both cases, if the rate of requests for this particular link is close to (or greater than) its capacity then the performances of the whole network will drop critically, although only a few number of stations will be overloaded.

Hence the idea of using the other routes that may exist between two nodes. The new problem that arises then is that, in distributed processing applications, the stations do not have a complete and instantaneous knowledge of the state of the system, and it is impossible to say, at any time, what is the best route towards a certain destination.

Recently, Mitra and Cieslak [10] described and analyzed a randomized routing algorithm in the case of Multistage Interconnection Networks. They obtained, in particular, that this system could complete communication requests in $O(LogN)$ where N is the number of inputs (and outputs) of the network. The constant in O depends on the maximal load of the links.

The aim of this paper is to analyze the efficiency of this method when the ordering of the packets has an importance, as in data transmission, voice communication or in distributed databases applications. As pointed out in Mitra's paper, the presence of several paths allows overtaking of packets, and induces an additional cost due to packet reordering. The main problem we deal with in the present paper is the evaluation of the effects of this resequencing time on the mean global response time of the system.

In part 1/, we describe the load balancing algorithm, and analyze it in the case of an asynchronous packet-switching network, with general Bernoulli Switches (with possibly uneven outcomes). We compute the mean "network response time" (the time needed to cross the network) and show how to choose the values of the switching probabilities in order to minimize it.

In part 2/, we study packet resequencing. We obtain, for a 1-extended network, an exact formula for the mean resequencing time of a packet. We also provide an asymptotic expansion when the number of stages in the network grows.

In part 3/, we use the formulas obtained in 1/ and 2/ to determine when the re-routing algorithm improves the performance of the system. Several examples of traffic distributions are considered.

Finally, in part 4/, we analyze the problem of adjusting the switching probabilities in order to minimize the response time of the network.

1/STOCHASTIC RE-ROUTING

A/Extension of the network

Our starting point is the n-stages Indirect Binary Cube, defined in [11], that we shall denote by: C_n^0 (Fig. 1). This network connects 2^n ($= N$) input wires to N output wires with n stages of $N/2$ switches each. We chosed this one for aesthetic reasons. As it is shown in [13] or [3], the five remaining "classical" networks (i.e. the Omega, Flip, Modified Data Manipulator, Baseline and Reverse Baseline networks) are topologically equivalent to this one: they can be obtained from C_n^0 by permuting the switches, column by column, in a convenient manner. Thus, all the results obtained in the following can be extended to them at the cost of a simple redrawing.

Let us first recall some basic and useful properties of this network.

- It has a right-recursive construction: it is obtained by superposing two $n-1$-stages Indirect Binary Cubes, and joining them by their right side with a n-th stage of switches, as shown in Fig. 2.
- It possesses the single path (also called "Banyan") property: there exists one single path connecting one given output, and one given output. An important consequence is that two paths that cross at one stage never cross elsewhere in the network, and similarly, there exists no third path that crosses these two fixed concurrent paths (Fig. 3).

The basis of the load sharing method is to add, before the first stage of the initial network, an extra stage called (1'), made of independent stochastic ("Bernoulli") switches that may send a packet, initially arriving at the input i, to the same input i with a certain probability, and to the input numbered $(i + 2^{n-1})$ *mod* N with the remaining probability. For the sake of simplicity, we choose to call p_i the probability that a packet will be send to the *upper* subnetwork (input i or $i - N/2$ according to the fact that $i < N/2$ or not), and $q_i = 1 - p_i$ (Fig. 4a).

Note that, modifying the labels of the inputs, we can draw this extra Bernoulli stage with "boxes", and see that the first stage (the connection between columns (1') and (1)) is a copy of the last one (between columns $n-1$ and n)(Fig. 4b). This shows also that Mitra's 1-extension of the Omega (Fig 5) and our 1-extension of the Cube are still topologically equivalent.

Using the two properties mentioned above (Banyan and Recursiveness), we see that packets going initially from input i to output j may now take two different and independent routes through the network, and join only after the last column (see Fig. 4 again). The fact that these two routes join only at the end of the network (just after the last switch) is essential for the following analysis.

B/Computation of the Network Response Time

In the case of packet switching communication, it is natural to model our network by a queueing network. We assume that every link (links are the critical resources) is in fact a ./M/1 server with mean service time $1/\mu$, infinite waiting room, and FIFO queueing discipline. These hypothesis are needed in order to use results of Queuing Networks Theory. The assumption on the service time distribution may possibly be weakened, using "insensitivity" results, but limiting constraints bearing on the size of the buffers would unfortunately forbid any simple analysis.

Customers that travel in the network are distributed in classes, according to their origin and destination. Suppose that class (i,j) customers arrive according to a Poisson Process of rate $\lambda_{i,j}$. We naturally assume that all services and input processes are independent, so that we have a Kelly-like open network, i.e. a queueing network with classes, deterministic routing (class by class), and class-independent queueing discipline [9].

Due to the FIFO discipline and the Banyan property, there is no possibility of overtaking between packets of the same class during their way through C_n^0. This allows to compute the

distribution of the sojourn time for an (i,j)-class customer in steady state. The details can be found in Appendix 1. One obtains that its Laplace transform is given by:

$$T^*_{i,j}(s) = \prod_{q \in P(i,j)} \frac{\mu - \hat{\lambda}(q)}{\mu - \hat{\lambda}(q) + s} \tag{1.1}$$

where:

- $P(i,j)$ is the set of queues that lie on the path from input i to output j
- $\hat{\lambda}(q)$ is the total rate of arrivals in queue q.

This gives the stability condition of the network, which reads:

$$\sup_q \hat{\lambda}(q) < \mu \,. \tag{1.2}$$

The mean sojourn time for a packet of class (i,j) is given by:

$$\bar{t}_{i,j} = \sum_{q \in P(i,j)} \frac{1}{\mu - \hat{\lambda}(q)} \,,$$

and the mean sojourn time of an arbitrary packet by (calling $\lambda = \sum \lambda_{i,j}$):

$$\bar{t} = \sum_{i,j} \frac{\lambda_{i,j}}{\lambda} \bar{t}_{i,j} = \frac{1}{\lambda} \sum_{i,j} \lambda_{i,j} \sum_{q \in P(i,j)} \frac{1}{\mu - \hat{\lambda}(q)} \,,$$

or, equivalently,

$$\lambda \bar{t} = \sum_q \frac{\hat{\lambda}(q)}{\mu - \hat{\lambda}(q)} \,. \tag{1.3}$$

Let us now compute the mean packet sojourn time in the extended network.

In addition to the new network response time, we should consider the time needed to pass the randomization part of the network. But we shall assume in all the following that the time needed to pass the Bernoulli stage is small with respect to the interarrivals, so that packets never interfere with one another. We shall hence assume that this "randomization time" is neglectible, and, in particular, that it does neither contribute to the phenomenon of desequencing, nor creates correlations between customers waiting times. We justify this hypothesis by the fact that a Bernoulli Switch performs no computations on the passing-by packets (in opposition with the 2×2 cells of the initial network), and just establishes a path. If the lengths of the packets are small, no interferences occur.

Therefore, we just have to rewrite (1.3) with the new values of the $\hat{\lambda}(q)$, that we denote by $\hat{\lambda}'(q)$. It is quite easy to see that if q is in the upper "sub-cube", or if it is in the n-th column, with an even label $(0, 2, \ldots)$, then

$$\hat{\lambda}'(q) = \sum_{q \in P(i,j)} p_i \lambda_{i,j} + p_{i+\frac{N}{2}} \lambda_{i+\frac{N}{2},j} \,,$$

and otherwise:

$$\hat{\lambda}'(q) = \sum_{q \in P(i,j)} p_i \lambda_{i,j} + p_{i-\frac{N}{2}} \lambda_{i-\frac{N}{2},j} \,.$$

But queues of a same column are naturally paired: customers of a certain class (i,j) may take two routes, that meet a given column in two queues: q and q^*. This two queues verify the following property: for all class (i',j') such that one path uses q, the other path will use q^*. Precisely, we have:

(r,c) is associated with $(r+N/2,c)$ if $1 \leq c \leq n-1$
(r,n) is associated with $(r+1,n)$ if r is even.

We may so rewrite (1.3), denoting by U the upper part of the network (i.e. the upper sub-cube, and the evenly labelled queues of the last column):

$$\lambda \bar{t} = \sum_U \frac{\hat{\lambda}'(q)}{\mu - \hat{\lambda}'(q)} + \frac{\hat{\lambda}'(q^*)}{\mu - \hat{\lambda}'(q^*)} . \tag{1.4}$$

Each of the terms in this sum is a convex function in the stability domain

$$D \triangleq \{(p_0,\ldots,p_{N-1}) \in [0,1]^N \mid \forall q,\ 0 \leq \hat{\lambda}'(q) < \mu \} ,$$

and has (if D is nonempty) an unique minimum when $\hat{\lambda}'(q) = \hat{\lambda}'(q^*)$.

From this, we conclude that $\bar{t}$ is minimum for a choice of the p_i's such that

$$\forall q \in U, \qquad \hat{\lambda}'(q) = \hat{\lambda}'(q^*) ,$$

or, equivalently, such that

$$\forall q \in U, \qquad \sum_{i,j \mid q \in P(i,j)} x_i \lambda_{i,j} + x_{i+\frac{N}{2}} \lambda_{i+\frac{N}{2},j} = 0 , \tag{1.5}$$

where we used the new variables $x_i = p_i - q_i$. Note that this is always verified when $x_i = 0 \quad \forall i$ i.e. when $p_i = \frac{1}{2} \quad \forall i$. The minimum is then :

$$\lambda \bar{t}_m = \sum_U \frac{\hat{\lambda}'(q) + \hat{\lambda}'(q^*)}{\mu - \frac{\hat{\lambda}'(q) + \hat{\lambda}'(q^*)}{2}} \tag{1.6}$$

All this computation is correct provided that

$$\forall q \in U, \quad \hat{\lambda}'(q) + \hat{\lambda}'(q^*) < 2\mu$$

that is:

$$\forall q \in U, \quad \hat{\lambda}(q) + \hat{\lambda}(q^*) < 2\mu \tag{1.7}$$

and this condition is weaker than (1.2), which proves that the capacity of the network is increased.

The system (1.5) translates the following statement: the best mean response time is obtained for a choice of the p_i's such that, for every class of customer, the two associated queues have the same global input rate (or load). We call this situation: "*Equi-Stream*".

As already noticed, Equi-Stream is reached when $p_i = q_i \quad \forall i$. It seems useless to allow the p_i's to take others values. That is right if we are concerned only with the network response time. But since we are interested also with the resequencing time (see 2/), it is interesting for *optimization* purposes to let p_i take other values (see 4/). This explains why we finish the study of (1.5) in the general case.

(1.5) is a linear homogeneous system with N variables, and a *priori*, $nN/2$ equations. In fact, this system is redundant, and, as shown in Appendix 3, it reduces to a system of $(n-1)N/4 + N/2$ equations. Then, two cases may appear:

- if $n = 1$ or 2, there are $N-1$ equations, and one parameter is free (at least). This case is very interesting, because it allows to control the resequencing time independently of the network response time. This will be discussed in section 4/.
- if $n \geq 3$, there are more equations than parameters, and, in the general case, the only solution of (1.5) is the null one. Note however that, under convenient constraints on the $(\lambda_{i,j})$'s, the system can be reduced sufficiently to free one parameter. For instance, if:

$$\forall i,j, \qquad \lambda_{i,j} = \pi_j \lambda_i \,. \tag{1.8}$$

This corresponds to the case where inputs demand outputs with the same probability π_j, but work at different rates λ_i. A very particular case of this one is the perfect equidistribution, with identical arrival rates at each input.

C/Recursive construction

We now consider a 1-extended cube (noted C_n^1), and apply the extension method to the upper and lower "sub-cubes". We introduce so a second Bernoulli Stage, (2'), between (1') and (1), which in turn may be represented with boxes (like in Fig. 5b) and appears to be a replica of the penultimate stage of C_n^0 (Fig. 6). So, due to the right-recursive structure of the Cube network, we can successively add Bernoulli stages, and construct the networks $C_n^1, C_n^2, \ldots, C_n^k, \ldots, C_n^n$. As we shall see, it is useless to add more Bernoulli columns. Correctly drawn, C_n^n is the concatenation of C_n^0 with its "mirror image". This network is *not* topologically equivalent to the concatenation of two Omega networks (which is the n-extension of the Omega by Mitra's construction), and looks like the Benes network ([2]). In fact, they would be equivalent if the two inner columns (n') and (1) were assimilated.

This progressive construction allows to verify that customers, initially of class (i,j), arriving in C_n^k, will in fact enter C_n^0 by 2^k ways, and follow 2^k different (although not fully disjoints) paths ending at output j. Paths who differ from the p-th Bernoulli stage will join again just after the $(n+1-p)$-th stage of C_n^0.

We shall now compute the switching probabilities that give a minimal response time in stationary behavior. The formulas given above and the convexity arguments still hold for each successive subnetwork. This probabilities are computed from the left to the right, because the successive values of the input rates in subnetworks are changed by the choices in the previous stages. Of course, setting all the probabilities $p_i^{(j)}$ to $\frac{1}{2}$ still leads to the optimal value of $\bar{t}$, but as in B/, this is not the only solution.

The research of the minimal response time of C_n^k thus leads to the resolution of k systems with N variables each. In fact:

1 system	with	N variables
2	with	$N/2$ variables
$\vdots$		$\vdots$
2^k	with	$N2^{-k}$ variables

In a fully extended network, the optimal time is reached when the switching probabilities are set to $\frac{1}{2}$ in the $n-2$ first randomization stages. For the rest, we have to solve $N/4$ systems of $3+2$ equations and $4+4$ variables, leaving each 3 free parameters. In the "identical demands" case (1.8), we have to solve successively n linear systems, of sizes $N-1, N/2-1, \ldots, N2^{-k}-1, \ldots, 1$,

that leave each one parameter free. The whole model has, in this case, n free parameters that may vary although the network keeps an optimal response time.

Let us now examine the stability conditions of C_n^k. We can see that the addition of the second random stage does no modify the arrival rates in the last column of C_n^0. For this column, the stability conditions in C_n^k are still given by (1.7).

This generalizes to the other columns: in stage p of C_n^0 (considered a part of C_n^k), queues are associated by groups, in the sense that they all belong to a possible route for a certain class (i,j). The number of queues in each group is 2^l, with

$$l \;=\; l(n,p,k) \;=\; \inf(k, n-p+1)\;.$$

When the re-routing probabilities are chosen as to provide a minimal mean response time, the total arrival rate in the queues of a same group will be equal. This situation is a "generalized Equi-Stream". Then, formula (1.6) becomes:

$$\lambda \bar{t}_m \;=\; \sum_{G group} \frac{l(G)\hat{\lambda}(G)}{l(G)\mu - \hat{\lambda}(G)}\;, \tag{1.9}$$

where

$$\hat{\lambda}(G) = \sum_{q \in G} \hat{\lambda}(q).$$

The stability condition is then given by:

$$\forall p,\ \forall G, \text{group in column } p,\quad \sum_{q \in G} \hat{\lambda}'(q) \;<\; \mu\, 2^l\;,$$

which, as for (1.7), is equivalent to

$$\forall p,\ \forall G, \text{group in column } p,\quad \sum_{q \in G} \hat{\lambda}(q) \;<\; \mu\, 2^l\;. \tag{1.10}$$

In particular, when $k = n$ and $p = 1$, this condition becomes

$$\sum_{i,j} \lambda_{i,j} \;<\; N\,\mu\;.$$

As expected, the maximum capacity of the system cannot exceed $N\mu$.

So, when n stages have been added, the Equi-Stream balances uniformly the initial streams over the N inputs of C_n^0. The only thing to do for improving further the performances of C_n^n would be to balance the streams in the other columns, beginning with the last one, which is the less balanced one (its stability conditions are the weakest ones). But this cannot be made adding more Bernoulli stages, because this cannot create more paths between one input and one output. That is why C_n^n may be called the "fully extended Cube".

This shows the principal limitation of the random re-routing method: as we do not change the final destination of the packets, it is not possible to balance the load of the links that are close to the outputs, as completely as for the distant ones.

2/RESEQUENCING TIME

In packet switching mode, the order of the packets is often of a great importance. This is the case in voice transmission, and generally in every application where a single data is split up in smaller packets, which are separately sent. This is also the case in a distributed computing/databases context: some sequences of computations/requests have to be executed in a precise order.

We already used that in the initially considered network, C_n^0, there was no risk of desequencing of packets, due to the FIFO discipline. This is not the case in extended networks, since packets of a given class may travel along several paths and overtake one another. So, if these packets have to be processed in the very order of their emission, some of them will have to experience, in addition to their network waiting time, an extra waiting time R, called resequencing time. R is the time a packet has to wait, before all the packets of his own class that have been emitted before him have reached the output link. This extra time appears as a cost to pay in counterpart of load balancing. Due to this negative effects on the performances, one can ask whether balancing the load actually decreases the mean communication time when including the resequencing time.

The resequencing problem is not new, and has been treated, for instance in [8] and [1]. But in this works, the disorder is created by the passage of customers through a system that creates identically distributed *independent* delays. This is not the case here, since waiting times of successive customers in a queue are correlated.

We shall consider the case $k = 1$, and show that, in steady state, we can compute exactly the expected value of R .

A/ The individual resequencing time

We shall compute the distribution of the resequencing time experienced by a customer of a given class.

Consider a "tagged" packet of class (i,j), arriving in the network. This packet will experience, along his path, a global waiting and service time A. When his last service has been completed, he has to wait until all the class (i,j)-packets that entered in the network before him have traversed the network. There are only two possible paths for the class (i,j)-packets, and no overtaking is possible along each of them. Thus, our tagged packet arrives after all the packets entered before him, that "chosed" the same way, and he has to wait only for the last packet entered before him that took the other way: when this one arrives, all the preceding ones will be arrived too. We suppose that the resequencing algorithm takes no time: packets leave the system as soon as the one that blocked them arrives. The case where this time is not zero has been considered in [1].

Let us call τ the time elapsed between the arrival of this last packet and our tagged one, whose arrival date is noted $t = 0$. Let also call B the delay experienced by this challenger packet. The exit dates of the two packets are respectively A and $B - \tau$, so that the resequencing time of our packet is (Fig. 7):

$$\begin{array}{rll} 0 & \text{if} & A \geq B - \tau \\ B - \tau - A & \text{if} & A \leq B - \tau \end{array}$$

which is classically written in the following form:

$$R = [B - (A + \tau)]^+ \tag{2.1}$$

where $[x]^+ \triangleq \max(0, x)$.

B/Independence of end-to-end delays

Our next task is to prove the independence of all considered random variables, when the network is of the type constructed above, and in stationary behavior. We need this to be able to compute the distribution of R from (2.1).

- A and (B,τ) are independent. This is a consequence of the recursive construction of C_n^0, for A and B depend only on the arrival dates and service demands of customers in the upper (resp. the lower) subnetwork. But the arrival processes in the two subnetworks are independent, because the result of the switching of a Poisson process by an independent Bernoulli toss is a couple of independent Poisson processes. And as these subnetworks are physically separated (there are no paths connecting them), no event occurring in one of them will ever have an influence on the history of the other. In particular, A and B are independent. From the independence of arrivals, we obtain that A and τ must also be independent.
- B and τ are independent. This would be clear if $t = 0$ was the date of an arrival in the lower subnetwork, as the arrival process is independent of the state of the network. In fact, due to the memoryless property of Poisson processes, we can do "as if" there were actually an arrival at this date. We prove this using the Marked Point Processes formalism ([6]).

Let us call $\mathbb{P}$ the point process of arrivals of class (i,j) customers, in the lower subnetwork, marked by their delay B. We want to compute $\mathbb{P}\{\tau \leq t,\ B \geq b\}$. We have the sequence of equalities :

$$\begin{aligned}
\mathbb{P}\{\tau \leq t,\ B \geq b\} &= \mathbb{P}\{-x_0 \leq t,\ k_0 \leq b\} && (2.2)\\
&= \mathbb{P}_0\{-x_1 \leq t,\ k_1 \leq b\} && (2.3)\\
&= \mathbb{P}_0\{x_1 \leq t,\ k_0 \leq b\} && (2.4)\\
&= \mathbb{P}_0\{x_1 \leq t\}\ \ \mathbb{P}_0\{k_0 \leq b\} && (2.5)\\
&= \mathbb{P}_0\{-x_{-1} \leq t\}\ \ \mathbb{P}_0\{k_{-1} \leq b\} && (2.6)\\
&= \mathbb{P}\{\tau \leq t\}\ \ \mathbb{P}\{B \leq b\}\,. && (2.7)
\end{aligned}$$

Equation (2.2) is the translation in terms of M.P.P.s : k_n is the mark carried by x_n. We pass from (2.2) to (2.3) using the fact that $\mathbb{P} = \mathbb{P}_0$ for a Poisson process, and for events that lie in the past. The save argument (reversed) leads from (2.6) to (2.7). To go from (2.3) to (2.4), and from (2.5) to (2.6), we use the fact that the process is stationnary, that is invariant for shifts. (2.4) is the expression of the independence of the arrival process. (2.7) proves the independence of τ and Q, Q.E.D.

C/Analytic computation of the resequencing time

Now, we can compute the distribution of R. As we do not know directly the distributions of A and B, but their Laplace transforms, we compute the Laplace transform of R: R^*. This can be done, reducing our problem to a "basic" boundary value problem [4]. We show here a way to compute $(\sup(B - A - \tau, 0))^*$, but the general problem of finding $(\sup(X,Y))^*$ can be treated in the same way.

Let us call $C = A + \tau$, $V = B - C$, B^* and C^* the Laplace Transforms of B and C:

$$B^*(s) = \int_{0^-}^{\infty} e^{-st}\, dB(t) \qquad \forall s\ \Re(s) \geq 0\,,$$

$$C^*(s) = \int_{0^-}^{\infty} e^{-st}\, dC(t) \qquad \forall\ \Re(s) \geq 0\,.$$

We want to compute the following integral, for $\Re(s) \geq 0$ (where $\Re(z)$ represents the real part of the complex z):

$$R^*(s) \;=\; \int_{0^-}^{\infty} e^{-st}\, dR(t) \;=\; \mathbb{P}\{R=0\} \;+\; \int_{0^+}^{\infty} e^{-st}\, dR(t) \tag{2.8}$$

But $V = B - C$, and B and C are independent, so:

$$V^*(s) \;=\; \int_{0^-}^{\infty} e^{-st}\, dV(t) \;=\; B^*(s)\, C^*(-s) \qquad \forall\, \Re(s) = 0$$

or:

$$\int_{-\infty}^{0^-} e^{-st} dV(t) \;+\; \mathbb{P}\{V=0\} \;+\; \int_{0^+}^{\infty} e^{-st} dV(t) \;=\; B^*(s) C^*(-s) \quad \forall\, \Re(s) = 0. \tag{2.9}$$

Equation (2.8) shows that the knowledge of R^* is related to the determination of the term

$$\int_{0^+}^{\infty} e^{-st}\, dV(t) \;\; .$$

The term $\mathbb{P}\{R=0\}$ is the obtained when s is set to 0 ($R^*(0) = 1$). Fortunately, (2.9) allows to compute this function, as the solution of a "basic" boundary value problem [4], that we shall recall briefly.

Let Φ be a complex valued function, regular in $\Re(z) > 0$ and $\Re(z) < 0$, vanishing when $z \to \infty$ (with $|\Re(z)| > \epsilon$ for some $\epsilon > 0$), continuous when approching $i\mathbb{R} = \{\Re(z) = 0\}$ in both $\{\Re(z) > 0\}$ and $\{\Re(z) < 0\}$. Call, for any $t \in i\mathbb{R}$:

$$\Phi^+(t) \;=\; \lim_{\substack{z \to t \\ \Re(z) < 0}} \Phi(z)\,, \Phi^-(t) \;=\; \lim_{\substack{z \to t \\ \Re(z) > 0}} \Phi(z)\,.$$

If, moreover, we want that

$$\Phi^+(t) \;-\; \Phi^-(t) \;=\; f(t) \quad \forall\, t,$$

for some complex function f, defined, continuous and Hölderian on $i\mathbb{R}$, then Φ is unique and its values are given by:

$$\Phi(z) \;=\; \frac{1}{2i\pi} \int_{i\mathbb{R}} \frac{f(t)}{t-z} dt\,, \quad \forall\, z \in \mathbb{C}\,. \tag{2.10}$$

This integral is to be taken in the principal value sense, if $t \in i\mathbb{R}$.

Returning to our problem, we just take:

$$\begin{aligned} \Phi(z) \;&=\; \int_{-\infty}^{0^-} e^{-xt}\, dV(t) && \text{for } \Re(z) < 0 \\ &=\; -\int_{0^+}^{+\infty} e^{-zt}\, dV(t) && \text{for } \Re(z) > 0 \end{aligned}$$

and we assume that dV has no mass concentrated on 0, so that $\mathbb{P}\{V=0\} = 0$. Our function Φ verifies then all needed assumptions. Now take $f(t) \;=\; B^*(t)\, C^*(-t)$ for $t \in i\mathbb{R}$. f is surely

continuous, and we shall assume, for the moment, that it verifies the Hölder condition (extended to infinity). Then (2.9) rewrites:

$$\Phi^+(t) \; - \; \Phi^-(t) \; = \; f(t) \quad \forall\, t \in i\mathbb{R}\,,$$

and by (2.10), we obtain:

$$\Phi(z) \; = \; \frac{1}{2i\pi} \int_{i\mathbb{R}} B^*(t)\, C^*(-t)\, \frac{dt}{t-z}\,,$$

and, using (2.8), we obtain the value of R^*:

$$R^*(s) \; = \; \mathbb{P}\{R=0\} \; - \; \Phi(s) \quad \forall\, \Re(s) \geq 0$$

Taking $s = 0$, we have

$$\mathbb{P}\{R=0\} \; = \; 1 \; + \; \Phi^-(0)$$

and finally,

$$R^*(s) \; = \; 1 \; + \; \Phi^-(0) \; - \; \Phi(s) \qquad \forall\, \Re(s) > 0\,. \tag{2.11}$$

This formula gives the distribution of $[B-C]^+$ (if $B \neq C$ *p.s.*, but it can be adapted to the other case), and has applications out of this study.

D/Application

The only missing thing to obtain R is the value of A and B. As τ and A are independent, C^* is easily computed from them:

$$C^*(s) \; = \; A^*(s)\, \tau^*(s) \qquad \forall\, \Re(s) \geq 0\,.$$

A is the total queuing and service time experienced by our tagged customer in stationary behaviour. As we already used it in 1/, its Laplace Transform is:

$$A^*(s) \; = \; \prod_{q \in P(i,j)} \frac{a(q)}{a(q)+s} \qquad \text{with } a(q) = \mu - \hat{\lambda}(q)$$

and, of course, B^* has a similar expression. If we call l the arrival rate of customers of class (i,j) on the other path than the tagged one's, we have:

$$\tau^*(s) \; = \; \frac{l}{l+s}$$

and finally, replacing in (2.11),

$$\Phi(z) \; = \; \frac{1}{2i\pi} \int_{i\mathbb{R}} \prod_{i=1}^{n} \frac{a_i}{a_i - t} \prod_{i=1}^{n} \frac{b_i}{b_i + t}\, \frac{\lambda}{\lambda - t}\, \frac{dt}{t-z} \tag{2.12}$$

where we replaced the $a(q)$ and $b(q)$ by (a_i) and (b_i), $1 \leq i \leq n$.

Note here that, in C/, we assumed that the function $B^*(s)C^*(-s)$ possessed the Hölder property. This is clearly the case if $n \geq 1$. Moreover, as A, B, τ have continuous distributions, $V \neq 0$ *p.s.*, and (2.11) applies.

Decomposing the integrand of (2.12) in elementary fractions, it is possible to integrate it, but the general formula ge get cannot be worked out to provide a more closed relation (but is suitable for numerical computations). As Φ must not possess any pole in $\{\Re(z) \geq 0\}$, we conclude that it is a rational function, with, as denominator, the product $\prod_{i=1}^{n} (z+a_i)(z+b_i)$, and, as numerator, a polynomial of degree n in (a_i) and (b_i) that must be symmetrical in (a_i)s and (b_i)s (but separately). The leading coefficient of this fraction is:

$$K = -\lambda \prod_{i=1}^{n} a_i\, b_i^{-1} \prod_{i,j} (a_i + b_j)^{-1} \prod_{i=1}^{n} (b_i + \lambda)^{-1} .$$

Unfortunately, we have not been able to find a simple expression for the numerator of Φ, despite its beautiful symmetry properties (see examples).

In application to our problem, we can now compute the mean resequencing time $\bar{r}_{i,j}$ for (i,j)-class customers. This is clearly:

$$\bar{r}_{i,j} = p_i\, \bar{r}_{i,j}^{\text{up}} + q_i\, \bar{r}_{i,j}^{\text{down}} \tag{2.13}$$

where $\bar{r}^{\text{up}}$ and $\bar{r}^{\text{down}}$ denote the queueing times for customers taking the upper and lower path respectively. Let $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ be the successive apparent service rates along the two paths. As the arrival rate of (i,j)-class packets in the lower path is $q_i \lambda_{i,j}$, we have:

$$\bar{r}_{i,j}^{\text{down}} = \bar{R}(a_1, \ldots, a_n, b_1, \ldots, b_n; q_i \lambda_{i,j}) \tag{2.14a}$$

and, exchanging up and down,

$$\bar{r}_{i,j}^{\text{up}} = \bar{R}(b_1, \ldots, b_n, a_1, \ldots, a_n; p_i \lambda_{i,j}) . \tag{2.14b}$$

The mean queueing time for all classes will be

$$\bar{r} = \sum_{i,j} \frac{\lambda_{i,j}}{\lambda}\, \bar{r}_{i,j} \tag{2.15}$$

and with formulas (2.11) to (2.14),replacing the a_is by their value $\mu - \hat{\lambda}(q_i)$, we obtain the (exact) general formula for $\bar{r}$ in C_n^0.

E/Examples

When $n \leq 2$, the calculation can be completely acheived, and gives:
For $n = 1$:

$$\Phi(z) = -\frac{ab\lambda}{(b+z)(a+b)(b+\lambda)} \tag{2.16}$$

$$R^*(z) = 1 - \frac{a\lambda}{(a+b)(b+\lambda)} \frac{z}{b+z}$$

$$\mathbb{P}\{R=0\} = R^*(\infty) = 1 - \frac{a\lambda}{(a+b)(b+\lambda)}$$

$$\bar{R} = -\frac{dR^*}{dz}(0) = \frac{a\lambda}{b(a+b)(b+\lambda)}$$

For $n = 2$:

$$\begin{aligned}\Phi(z) \;=\; K[\; & b_2^4 - b_1^4 \\ & + (b_2^3 - b_1^3)(a_1 + a_2 + \lambda + z) \\ & + (b_2^2 - b_1^2)(a_1a_2 + a_1\lambda + a_1z + a_2\lambda + a_2z + \lambda z) \\ & + (b_2 - b_1)(a_1a_2\lambda + a_1a_2z + a_1z\lambda + a_2\lambda z)]/(b_2 - b_1)\end{aligned}$$

$$\bar{R} \;=\; \lambda \prod_{i=1}^{n} \frac{a_i}{b_i} \prod_{i,j} (a_i + b_j)^{-1} \prod_{i=1}^{n} (b_i + \lambda)^{-1} (b_2 - b_1)^{-1}$$
$$[b_2^5 - b_1^5 + (b_2^4 - b_1^4)(a_1 + a_2 + \lambda) + (b_2^3 - b_1^3)(a_1a_2 + a_1\lambda + a_2\lambda) + (b_2^2 - b_1^2)(a_1a_2\lambda)].$$

For $n \geq 3$, computation "by hand" becomes really hard, except in the special case where all apparent service rates are equal. Simplifications in the formula lead then to a closed formula for $\bar{R}$, and to an asymptotic analysis (when the number of queues grows). We shall present here these results : they will be useful in 3/.

F/Asymptotic expansion of $\bar{R}$

We study the network C_n^1 in "perfect balance", i.e. when all the arrival rates $\lambda_{i,j}$ are equal , and all routing probabilities are $\frac{1}{2}$. An arriving customer sees it as two identical serie-queueing networks of length n, in parallel (Fig. 8). We shall set, for convenience $\lambda_{i,j} = 2\lambda \; \forall i, j$. The network behaves as if customers of each class arrived independently along two inputs, according to Poisson processes of same intensity λ.

We are interested in the behavior of the mean resequencing time, $\bar{r}_n$ when $n \to \infty$. We shall find the leading term in the expansion of $\bar{r}_n$, using the Central Limit Theorem.

Using the results of 1/, and the notations of 2/, we see that the distributions of A_n and B_n are the same as the sum of n independent r.v. exponentially distributed with mean $a = \mu - \lambda N/2$. The Central Limit Theorem gives then:

$$\frac{A_n - n/a}{\sqrt{n}/a} \to \mathcal{N}(0,1) \;, \qquad n \to \infty \tag{2.17}$$

and the same thing for B_n. Since the opposite of a normally distributed r.v. is still normally distributed,

$$-A_n \to \mathcal{N}(\frac{n}{a}, \frac{\sqrt{n}}{a}) \;.$$

All the present r.v.s being independent, we have, for the mean resequensing time:

$$\bar{R}_n = E([B_n - (A_n + \tau)]^+) \;=\; E([X_n - \tau]^+),$$

where

$$X_n \triangleq B_n - A_n \to \mathcal{N}(0, \frac{\sqrt{2n}}{a}) \;.$$

This quantity rewrites, conditioning by τ :

$$E([X_n - \tau]^+) \;=\; E([X_n]^+) \;-\; \frac{1 - X_n^{+*}(\lambda)}{\lambda}$$

where X_n^{+*} is the Laplace transform of $[X_n]^+$. As $|X_n| \to \infty$ when $n \to \infty$, we can compute that $X_n^{+*}(x) \to 1/2$ for all finite x. So, $E([X_n - \tau]^+)$ tends to $E([X_n]^+) - 1/2\lambda$, as $n \to \infty$. A simple computation gives then:

$$\bar{R}_n \sim \frac{1}{a}\sqrt{\frac{n}{\pi}} \tag{2.18}$$

To obtain the following terms, we use the formula obtained in 2/. Here, (2.12) writes:

$$\Phi_n(z) = \frac{1}{2i\pi}\int_{\mathrm{IR}} \left(\frac{a^2}{a^2+t^2}\right)^n \frac{\lambda dt}{(\lambda - it)(it - z)} .$$

After some technical computations, due to the fact that we manipulate integrals that are taken in the Principal Value sense, we obtain that

$$\bar{R}_n = \frac{1}{a}\sqrt{\frac{n}{\pi}} - \frac{1}{2\lambda} + (\frac{a}{2\lambda^2} - \frac{1}{8a})\frac{1}{\sqrt{\pi n}} + \frac{1}{\sqrt{n}}\, o(1) . \tag{2.19}$$

In conclusion of this section, we point out that, if the *load* of the network is held constant, then the mean resequencing time in in $O(\sqrt{n})$, and so tends to disappear in comparison to the mean network delay, which (from (1.3)) is in $O(n)$.

G/Resequencing time for larger extensions

The computation we presented above for C_n^1 does not apply to larger extensions. Formula (2.1) is still true, but now, the random variables A and B are no longer independent, because customers may take some paths that have a common part. When they pass in a same queuing line, their network response time become dependent. It seems that no exact analytic solution can be obtained.

However, there is hope that some bounds may be obtained, using "Stochastic Ordering" techniques ([12]).

Finally, we mention an interesing problem that arises here: is it more interesting to resequence customers at the end of the network, or as soon as possible (when two paths join). Some recent results seem to prove that the resequencing "as late as possible" gives best results. Anyway, one can ask if it is worth fitting out each switching cell with a packet-reordering mechanism.

3/RESULTS

In this section, we apply the results of 2/. We consider the 1-extended cube, and fix all the switching probabilities to $\frac{1}{2}$. We suppose that the time needed to traverse the randomization stage is neglectible.

We compute, for various values of the input rates $\lambda_{i,j}$, the mean global waiting time for an arbitrary customer in the initial network C_n^0 (with (1.3)) and in C_n^1 (with (1.6) and (2.11) to (2.15)). We obtain the value or the gain due to the method: g, which is:

$$g = \bar{t}_{init} - \bar{t}_m - \bar{r} . \tag{3.1}$$

Using (1.3) and (1.6) we can compute

$$\Delta\bar{t} = \bar{t}_{init} - \bar{t}_m = \frac{1}{\lambda} \sum_U \frac{(\hat{\lambda}(q) - \hat{\lambda}(q^*))^2}{(\mu - \lambda(q))(\mu - \lambda(q^*))(2\mu - (\lambda(q) + \lambda(q^*))} \tag{3.2}$$

From (3.2), we see that two quantities may have a great influence on the value of g: the maximum load of the queues in the initial network, but also the sum $\sum(\lambda(q) - \lambda(q^*))^2$ which can be seen as the variation of the network with respect to the Equi-Stream state.

A/The Almost Balanced Network

In this experience, we suppose that

$$\lambda_{i,j} = \lambda \qquad \forall (i,j) \neq (0,0).$$

The network is almost balanced, the only lack of balance comming from the $(0,0)$-class customers. In this case, one can foresee that the gain of the re-routing method is not necessary positive. In the worst case, when $\lambda_{0,0} = \lambda$, the network is already in the Equi-Stream state, and the gain of network traversing time if null. As the resequencing time is positive, the gain is negative... Therefore, for values of $\lambda_{0,0}$ in the neighbourhood of λ, g will still be negative.

We drawn in Fig 9 the curves $g_n(N\lambda, \lambda_{0,0}) = 0$ for different values of n. The abscissas represent the load of all the queues that are not in $P(0,0)$ (see in 1/B/ the definition of P). The size of the network goes from $n = 1$ to 5. The continuous straight lines represent the limits of the stability domain of C_n^1, and the dashed ones those of C_n^0.

Some remarks from this diagram:

- When $n = 1$ or $n = 2$, there are some values of $(\lambda, \lambda_{0,0})$ with $\lambda > \lambda_{0,0}$ that give a positive gain. When $n \geq 3$, the "hole" created in the queues of $P(0,0)$ by the lack of $(0,0)$-customers is not deep enough: the gain $\Delta\bar{t}$ of queueuing time can not balance the resequencing time.
- When n grows, the curve $g_n = 0$ tends (slowly) to zero. This can be explained by the fact that if we fix $(N\lambda, \lambda_{0,0})$, then $\Delta\bar{t}$ grows in $\mathcal{O}(n)$, and $\bar{R}$ grows like $\mathcal{O}(\sqrt{n})$ (from the results of 2/F/). Therefore, their difference becomes positive when n tends to infinity. However, if we fix the load $N\lambda$ and the ratio $\lambda/\lambda_{0,0}$, the situation changes: $\Delta\bar{t}$ goes to zero, and $\bar{R}$ tends to a finite value.

In conclusion, Fig 9 shows that in the case where the traffic of $(0,0)$ customers is somewhat greater than those of the rest of the network, the re-routing algorithm will improve the performances of the network. This is true *a fortiori* if the network is not in perfect balance.

Note also that we fixed all the p_i's to $\frac{1}{2}$, in order to be consistent with Mitra's model. The best thing to do is, of course, to set $p_i = 0 \quad \forall i \geq 1$, and to find the optimal value for p_0. This optimal value is not easy to compute, but taking $p_0 = \frac{1}{2}$ leads to quite good results.

B/Stochastically balanced networks

Here, we suppose that the arrival rates $\lambda_{i,j}$ are random variables, independent and identically distributed according to a law F. Thus, all the queues of the network have the same expected load, but each sample provides a different distribution of this load.

We let the size of the network vary between 4 and 7, and took for F five different distributions with same mean $3/4N$. These distributions are classified in order of increasing variance in table 1 below. For each couple (size,distribution), we generated 100 samples of traffic intensities, and computed the value of g. According to the remark concerning equation (3.2), we chosed to plot the results in the (λ_{max}, g)-plane, where λ_{max} is the maximum value of $\lambda(q)$ in the initial network.

The result of each experiment is a cloud of points. In Fig. 10a and Fig. 10b, we superimposed the clouds obtained for the five distributions n being fixed, to 6 and 7 respectively. In Fig. 11a and Fig. 11b, we fixed the law and let n vary. We represented respectively d_2 and u_2.

From Fig. 10, we see that the five clouds are roughly arranged along a climbing line, in increasing order of variance. This is clear fo u_1, d_3 and even d_1. This is less clear for d_2 and u_2, what can ba explained by the fact that these two distributions have almost the same variance. Points of the u_2-cloud tend to have a greater λ_{max}, but the gains are comparable. Note that when $n = 7$, the clouds are more concentrated, and separated from each other. Note also that u_1 gives always bad results and that d_3 gives always good ones. d_1 is "mostly" bad, and the two remaining mostly good.

From Fig. 11, we see the effect of increasing the size of the network: as n grows, λ_{max} goes to $\frac{3}{4}$, and g goes to 0. This, in fact, is a consequence of the Law of Large Numbers: when N grows, the network tends to be in perfect balance, with same load in all queues. As well, we can show that λ_{max} actually tends to $\frac{3}{4}$ also. We can say that these networks are asymptotically balanced. Distributions with greater variance give networks with a greater disorder, and a greater maximum load, which explains the results of Fig. 10.

Finally, note that Fig11. indicates that, for each n, there might exist some limiting variance (here between 1/16 and 1/12) which would give a null gain, on the average. The gain would be negative (or mostly negative) for smaller variances and positive for bigger ones.

Name	Description	σ^2	$\sigma^2/\sigma^2(u_1)$	symbol
u_1	Uniform on $[\frac{1}{2N}, \frac{1}{N}]$	$\frac{1}{48}$	1	o
d_1	Discrete on $(\frac{1}{2N}, \frac{1}{N})$ weights $(\frac{1}{2}, \frac{1}{2})$	$\frac{1}{16}$	3	-
u_2	Uniform on $[\frac{1}{4N}, \frac{5}{4N}]$	$\frac{1}{12}$	4	^
d_2	Discrete on $(0, \frac{3}{4N}, \frac{1}{N})$ weights $(\frac{1}{8}, \frac{1}{2}, \frac{3}{8})$	$\frac{3}{32}$	4.5	+
d_3	Discrete on $(0, \frac{1}{N})$ weights $(\frac{1}{4}, \frac{3}{4})$	$\frac{3}{16}$	9	*

Table 1

C/CONCLUSIONS

From the examples studied above, we conclude that, there are two quantities that rule the value of this gain: the variance of the load distribution, and the maximal value of the load (which depends on both mean and variance of the load distribution). Even when the network is almost balanced, a small local disorder suffices to make the load sharing advantageous.

Therefore, we conclude that, unless the load of the network is supposed to be very small or particularly balanced, random load balancing is a good idea. Now, as well in the case of massive parallel computing as in the case of telephone switching, the hypothesis of a perfectly balanced

load is difficult to explain: it seems that, even if the rate of request at the input were equal, the distribution of demands over the outputs would not be uniform, but rather concentrated over a few number of privileged outputs (possibly changing over long periods of time). In this case, the interest of the method is obvious. We intend to perform specific computations for this case.

4/OPTIMIZATION

As we pointed out in section 1/, there are values of the size of the network and/or of the input rates such that there is not an unique way of sharing the load: there might be one or several parameters free, that can vary without changing the value of the mean network response time. The point is that the mean resequencing time *does* depend on the values of the p_i's and has no reason to remain constant when these probabilities change in the Equi-Stream space (or in a parallel space, as we shall see). To get convinced of this, one may remark that $\bar{r}_{i,j}$ is null for $p_i \in \{0,1\}$, and strictly positive otherwise.

We wish therefore use this interesting freedom to improve again the performances of the network. Our goal is to obtain the best possible configuration, in the sense that the mean global response time, $\bar{t} + \bar{r}$ (all classes together) will be minimized (the values of the arrival rates being fixed) . We mention here the even more difficult problem of minimizing this time when the time spent in the Bernoulli Stages is not neglectible, but depends of the number of stages added. It seems however that, considering a physical realization of the system, it may not be realistic to allow the arriving customers to jump over some of the Bernoulli Stages.

In the sequel, we shall restrict ourselves to the study of C_n^1, the only case where we know the value of the mean resequencing time. The problem is, given the set $\{\lambda_i, j,\ 0 \leq i,j < N\}$, to find the values of $\{p_i, 0 \leq i < N\}$ that minimize the function $\bar{t}+\bar{r}$. Considering the number of variables, there are chances that the solution of the problem is analytically untractable. On can, however, state some general remarks.

A/Mininization of $\bar{t} + \bar{r}$ in C_n^1

We have to study the behavior of $\bar{t} + \bar{r}$ on the domain D, defined in 1/2/:

$$D = [0,1]^N \cap \{\vec{p} \in \mathbb{R}^N, \mid \forall q,\ \hat{\lambda}'(q) < \mu\} ,$$

or, equivalently, using $x_i = p_i - q_i = 2p_i - 1$, on

$$D' \triangleq [-1,1]^N \cap \{\vec{p} \in \mathbb{R}^N, \mid \forall q,\ \hat{\lambda}'(q) < \mu\} .$$

First, some geometric properties of D'. As the queues are paired, as $\hat{\lambda}'(q) + \hat{\lambda}'(q) = \hat{\lambda}(q^*) + \hat{\lambda}(q^*)$, $\forall q$, and as the fact of changing for every i, x_i to $-x_i$ (or p_i to q_i) is equivalent to exchanging q and q^*, we have:

The closure of D' is a convex polyhedron, with $\vec{0}$ as center of symmetry.

Its faces are either those of the probability cube (these faces are in D'), either given by:

$$\{\hat{\lambda}'(q) = \mu\} \quad \text{or} \quad \{\hat{\lambda}'(q^*) = \mu\},$$

for some $q \in U$, in which case the face does not belong to D' ("unstable face").

Consider the case where, at the Equi-Stream, one variable p_i is free. As it varies, the other p_j vary with it, and the point $\vec{p}$ moves along a straight line (the Equi-Stream line) which is parallel to every unstable face of D', and contains $\vec{0}$. If more than one variable is free, p describes an Equi-Stream space whose dimension is the number of free p_js. Along each line parallel to the Equi-Stream line, all the $\hat{\lambda}'(q)$ keep a constant value, so that the distribution of the network response time along each path is constant (and also $\bar{t}$). Applying the results of Appendix 2, we have:

for all i and j, $\bar{r}_{i,j}$ is a concave function of p_i

and therefore, using (2.15):

$\bar{r}$ is concave on any segment of D' parallel to the Equi-Stream line, and is therefore minimal at one edge of this segment.

So, the minimum we search is somewhere in a stable face of D'. The dimension of the problem is reduced by one. But if this dimension is big, there is still work to do, for, unfortunately, the same argument cannot be applied recursively.

In the other case, when the Equi-Stream space is reduced to $\{\vec{0}\}$, we don't have any precision about the location of this minimum. Our conjecture, justified by the results of part 3/, is that, provided that the network is not particularly balanced, the minimum is not located "very far" from the Equi-Stream point.

The Case $n = 1$

This case is a multi-class version of the network costituted of 2 identical ./M/1 servers in parallel, where arriving customers can be sent to either of the servers according to a Bernoulli Process which rate depends on the classes alone, and where exitting customers are to be resequenced, class by class. In this case, the dimension of the optimization problem reduces to 1, and we are able to derive explicitly the complete formulae, and obtain the solution. This work is done in [7].

We just mention here that, except in special cases, the optimal switching probability is given by a root of a fifth degree polynomial, and therefore has no closed expression.

C/The General Case

When $n > 1$ (and even for $n = 2$), the number of variables forbids any analytical solution. Only numerical solutions (using the formulae obtained in 2/) can be obtained.

This is not fully satisfying, if we want to control dynamically the system, computing the p_i, on the basis of estimates of the traffic intensities, which would change frequently. That is why we think that the Equi-Stream value might be a good choice because it is relatively easy to compute, and even trivial the natural solution $\{\vec{x} = \vec{0}\}$ is satisfactory, and because "in most cases", it provides an improvement in the global response time. Thus, the study should be oriented towards the characterization of the region called "nearly balanced region", in order to decide in which measure this assertion is justified.

D/The Multi-Extended Cube

The study of $C_n^k, k > 1$ if, for the moment, almost completely unexplored. Mean value results can be obtained for the computation of the network response time, and we refer to [10] for an worst case analysis.

As for the value or the resequencing time, see remark in 2/G/. No doubt that some deeper study should be made in this way.

APPENDIX 1

We intend to prove that, in the C_n^0 network described in I/, with multiple classes of customers and deterministic routing, we have:

Theorem

Let T_k, $k = (i,j)$ the random variable representing the time spent in the network by a customer of class k, in stationnary behavior. The Laplace Transform of the end-to-end delay is of the following form:

$$T_k^*(s) = \prod_{q \in P(k)} \frac{a(q)}{a(q)+s}, \quad \forall \Re(s) \geq 0 \,. \tag{A1.1}$$

We assumed in 1/B/ that all input processes were independent Poisson processes, that all the queues were ./M/1 independent servers with infinite waiting room and FIFO queuing discipline. Every class k possesses a fixed route $P(k)$. Under these conditions, C_n^0 is an open Kelly Network, and we can apply the results of [9]. In particular, we have:

Proposition 1

If: Q is the set of queues in C_n^0,
K is the number of classes,
$n(q)$ is the number of customers in q, among which $n_k(q)$ are of class k,
$\hat{\lambda}(q)$ is, ad defined in 1/b/, the global rate of arrivals in queue q, and $\hat{\lambda}_k(q)$ the partial rate for the class k,
$\underline{n}$ represents a particular distribution of customers in the network,
then:

$$\mathbb{P}(\underline{n}) = \prod_{q \in Q} \left(1 - \frac{\hat{\lambda}(q)}{\mu}\right) n(q)! \prod_{k=1}^{K} \frac{1}{n_k(q)!} \left(\frac{\hat{\lambda}_k(q)}{\mu}\right)^{n_k(q)} \tag{A1.2}$$

As only N different classes of customers may enter a given queue q, the inner product in (A1.2) simplifies to:

$$\prod_{k|q \in P(k)} \frac{1}{n_k(q)!} \left(\frac{\hat{\lambda}_k(q)}{\mu}\right)^{n_k(q)} .$$

By a classical sommation, the equation (A1.2) leads to the micro-state probabilities for queue q:

$$\mathbb{P}(n(q)) = \left(1 - \frac{\hat{\lambda}(q)}{\mu}\right) n(q)! \prod_{k|q \in P(k)} \frac{1}{n_k(q)!} \left(\frac{\hat{\lambda}_k(q)}{\mu}\right)^{n_k(q)} . \tag{A1.3}$$

which justifies the name of "Product Form" given to (A1.2).

Now, we shall compute the generating function of the number of customers of class k in the whole network. We have, for all $|z| < 1$:

$$N_k(z) \triangleq \sum_0^\infty \mathbb{P}(n_k = p) z^p$$
$$= \sum_{p=0}^\infty \prod_{\sum n_q = p} \mathbb{P}(n_k(q) = n_q)\ z^{n_q}$$

and so, if $N_k(q)$ is the generating function of the number of packets of class k in queue q, we have:

$$N_k(z) = \prod_{q \in P(k)} N_k(q)(z)\ . \qquad (A1.4)$$

But, using (A1.3), we can compute:

$$\mathbb{P}(n_k(q) = p) = \left(1 - \frac{\hat\lambda(q)}{\mu}\right) \frac{1}{p!} \left(\frac{\hat\lambda_k(q)}{\mu}\right)^p \sum_{q=0}^\infty (p+q)! \sum_{\substack{(n_i)\\ i \in P(k)-k \\ \sum n_i = q}} \prod_{i \mid i \in P(k) \atop i \neq k} \frac{1}{n_i!} \left(\frac{\hat\lambda_i(q)}{\mu}\right)^{n_i}$$

$$= \left(1 - \frac{\hat\lambda(q)}{\mu}\right) \left(\frac{\hat\lambda_k(q)}{\mu}\right)^p \sum_{q=0}^\infty \frac{(p+q)!}{p!q!} \left(\frac{1}{\mu} \sum_{\substack{i \in P(k) \\ i \neq k}} \hat\lambda_i(q)\right)^q$$

$$= \left(1 - \frac{\hat\lambda(q)}{\mu}\right) \left(\frac{\hat\lambda_k(q)}{\mu}\right)^p \sum_{q=0}^\infty \frac{(p+q)!}{p!q!} \left(\frac{\hat\lambda(q) - \hat\lambda_k(q)}{\mu}\right)^q , \qquad (A1.5)$$

so that:

$$N_k(q)(z) = \left(1 - \frac{\hat\lambda(q)}{\mu}\right) \sum_{p=0}^\infty \sum_{q=0}^\infty \frac{(p+q)!}{p!q!} \left(\frac{z\hat\lambda_k(q)}{\mu}\right)^p \left(\frac{\hat\lambda(q) - \hat\lambda_k(q)}{\mu}\right)^q$$

$$= \frac{\mu - \hat\lambda(q)}{\mu - \hat\lambda(q) + \hat\lambda_k(q)(1-z)}\ . \qquad (A1.6)$$

In stationnary behavior, if $D_k(z)$ is the generating function of the number of class k customers at a class k departure instant, we have:

$$N_k(z) = D_k(z)\ . \qquad (A1.7)$$

Let us compute D_k. As there is no overtakig, the number of k-customers left in the network by a tagged customer is the number of k-customers arrived while this customer was in the network: a_k. Conditionning by a_k and T_k:

$$D_k(z) = \sum_{p=0}^\infty \int_0^\infty z^p\ \mathbb{P}(a_k = p \mid T_k = t)\ dT_k(t)\ .$$

But, because of the non-overtaking and the single path properties, we know that T_k does not depend, neither directly nor indirectly, on the story of the k-customers entered after him: our

tagged customer will never meet these ones, and will never meet others, influenced by them. Thus, we can write:

$$\mathbb{P}(a_k = p \mid T_k = t) \;=\; \mathbb{P}(p \text{ arrivals in } [0,t[) \;,$$

and, as the arrival process is a Poisson process,

$$\begin{aligned} D_k(z) \;&=\; \int_0^\infty e^{-\hat{\lambda}_k(q)tz} dT_k(t) \\ &=\; T^*(\hat{\lambda}_k(q)(1-z)) \end{aligned} \qquad (A1.8)$$

Now, we just have to replace (A1.6) in (A1.8), to obtain:

$$T_k^*(\hat{\lambda}_k(1-z)) \;=\; \prod_{k|q\in P(k)} \frac{\mu - \hat{\lambda}(q)}{\mu - \hat{\lambda}(q) + \hat{\lambda}_k(q)(1-z)} \;,$$

which is (A1.1), taking $s = \hat{\lambda}_k(1-z)$, $a(q) = \mu - \hat{\lambda}$, and modulo an analytic continuation of T_k^* on the whole domain $\{\Re(z) \geq 0\}$.

APPENDIX 2

We prove here the concavity of the reseqencing time in the case of a constant disordering environment.

Consider a system of two independent subsystems in parallel. Customers arrive according to a Poisson process, switched according to an independent Bernoulli process of parameter p. We keep the notations of 2/C/, but we dont assume A and B to be exponentially distributed anymore. We assume instead that these random variables have a law independent of the value of p. If $\bar{r}$ is the expected resequencing time of an arriving customer, then we have:

Theorem

$\bar{r}$ is a concave function of p.

Proof :

As arriving customers are scattered over the two queues, we must replace (2.1) by:

$$R = p\,[B - (r + A)] + q\,[A' - (r' + B')]\,, \qquad (A2.1)$$

where A', B' and r' play the same role than A, B, r for the customers that have been routed to the lower subsystem. A and A', and B and B' are identically distributed. r and r' are still exponentially distributed, but with respective rates: ql and pl.

Let $C = [B - A]^+$. Conditionning by the value of C, we have:

$$\begin{aligned}
\mathbb{E}([B - (r + A)]^+) &= \mathbb{E}\left(\int_0^C qle^{-ql}(C - t)dt\right) \\
&= \mathbb{E}\left([-e^{-qlt}(C - t)]_0^C - \int_0^C e^{-qlt}dt\right) \\
&= \mathbb{E}(C) - \mathbb{E}\left(\frac{1 - e^{-qlC}}{ql}\right) \\
&= \mathbb{E}(C) - \frac{1 - C^*(ql)}{ql}\,.
\end{aligned}$$

Introducing $D = [A - B]^+$, we obtain:

$$\bar{r} = pE(C) + qE(D) - \left(p\frac{1 - C^*(ql)}{ql} + q\frac{1 - D^*(pl)}{pl}\right)\,. \qquad (A2.2)$$

The functions

$$G(x) = \frac{1 - C^*(x)}{x} \quad \text{and} \quad H(x) = \frac{1 - D^*(x)}{x}$$

are convex and decreasing with respect to x. This is because C^* and D^* are Laplace transforms of probability distributions, and therefore are completely monotone (see [FEL] for details). Then $G(lq)$ is convex and increasing in p (remember that $q = 1 - p$). As the product of two positive convex identically monotone functions (both increasing or decreasing) is convex, then $pG(qp)$ and

$qH(pl)$ are convex functions of p, and so is the last term of (A2.2). Its opposite is then concave, and as the linear term in (A2.2) can be considered as concave, Theorem 1 is proved.

Corollary :

If A and B have the same distribution, then $\bar{r}$ has a single maximum when $p = q = \frac{1}{2}$.

Proof :

If A and B have the same distribution, so do C and D. Then $\bar{r}$ is symmetrical in p and q, and has an extremum in $p = q$. It is a maximum by concavity, and it is unique, still by concavity.

APPENDIX 3

We prove that the Equi-Stream system (1.5) reduces from $nN/2$ to $N/2 + N/4(n-1)$ equations.

First, remark that C_n^0 possesses the so-called "Buddy Property" ([3]), that is: the connections between two consecutive columns reduce to the simple network in Fig A1.

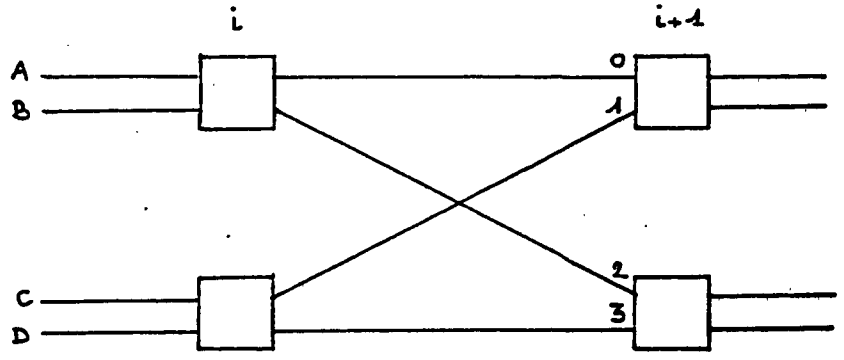

Fig A1 A "Buddy Pair"

Note then that, as no packets are lost, we have the conservation equations:

$$\begin{cases} \hat{\lambda}(0) + \hat{\lambda}(2) &= \hat{\lambda}(A) + \hat{\lambda}(B) \\ \hat{\lambda}(1) + \hat{\lambda}(3) &= \hat{\lambda}(C) + \hat{\lambda}(D) \end{cases}.$$

Suppose now we have written the Equi-Stream equalities up to column i. Writing them for the $i+1$-th column, we must have:

$$\begin{cases} \hat{\lambda}(0) = \hat{\lambda}(0^*) \\ \hat{\lambda}(1) = \hat{\lambda}(1^*) \end{cases}.$$

But then,

$$\begin{aligned} \hat{\lambda}(2) &= \hat{\lambda}(A) + \hat{\lambda}(B) - \hat{\lambda}(0) \\ &= \hat{\lambda}(A^*) + \hat{\lambda}(B^*) - \hat{\lambda}(0^*) \\ &= \hat{\lambda}(2^*) , \end{aligned}$$

and, for the same reason,

$$\hat{\lambda}(3) = \hat{\lambda}(3^*) .$$

Thus, the equations written for the i-th column, joined with half the equations written for the $i+1$-th column imply the other half. In the initial system, $1/2 \times N/2 \times (n-1)$ equations are redundant, QED.

REFERENCES

[1] BACCELLI F., GELENBE E. and PLATEAU B. An End-to-End Approach to the Resequencing Problem. *JACM, Vol. 31, No. 3, July 1984, pp. 474-485.*

[2] BENES, V.E. *Mathematical Theory of Connecting Networks and Telephone Traffic.* N.Y. Academy, 1965.

[3] BERMOND J.C, FOURNEAU J.M, JEAN-MARIE A. A Graph Theoretical Approach to Equivalence of Multi-Stage Interconnection Networks. To appear in the *Information Processing Letters.*

[4] COHEN J.W., BOXMA O.J. *Boundary Value Problems.* North Holland 1982

[5] FELLER W. *An Introduction to Probability Theory and its Applications.* J. Wiley & Sons, 1971.

[6] FRANKEN P et al. *Queues and Point Processes.* Springer Verlag 1981.

[7] JEAN-MARIE A. Load Balancing in a System of Two Queues with Resequencing. INRIA report, to appear.

[8] KAMOUN F., KLEINROCK L. and MUNTZ R. Queuing Analysis of the Ordering Issue in a Distributed Database Concurrency Control Mechanism. *Proc. 2nd Intl. Conf. on Distr. Comp. Systems, Versailles, France, Apr. 1981.*

[9] KELLY F.P. *Reversibility and Stochastic Networks.* J. Wiley & Sons, 1979.

[10] MITRA D. and CIESLAK R. Randomized Parallel Communications on an Extension of the Omega Network. *Proc. of the 1986 Intl. Conf. on Parallel Processing.*

[11] PEASE M.C. The Indirect Binary Cube Microprocessors Array. *IEEE Trans. Comp. Vol C26, pp. 458-473, May 1977.*

[12] STOYAN D. *Comparison Methods for Queues and Other Stochastic Models.* J. Wiley & Sons 1983.

[13] WU C. and FENG T. On a Class of Multistage Interconnection Networks. *IEEE Trans. Comp. Vol. C29, pp 694-702, Aug. 1980.*

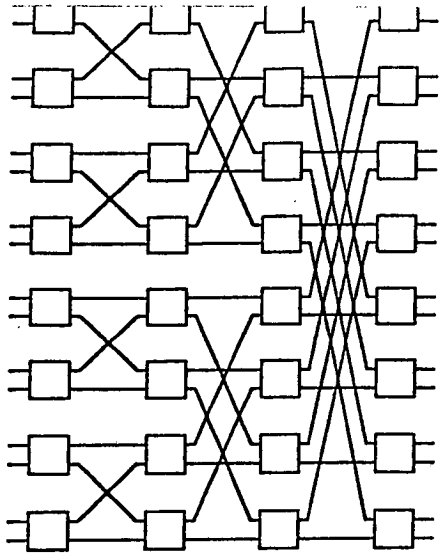

Fig 1 16x16 Indirect Binary Cube

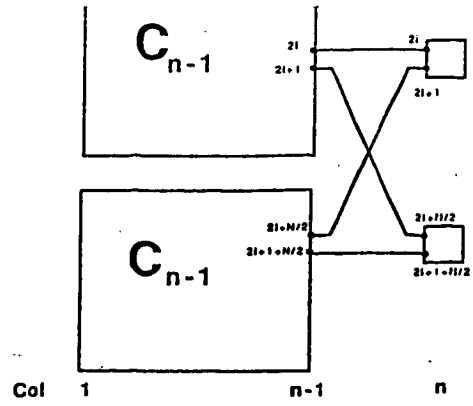

Fig 2 Recursive Construction of C_n

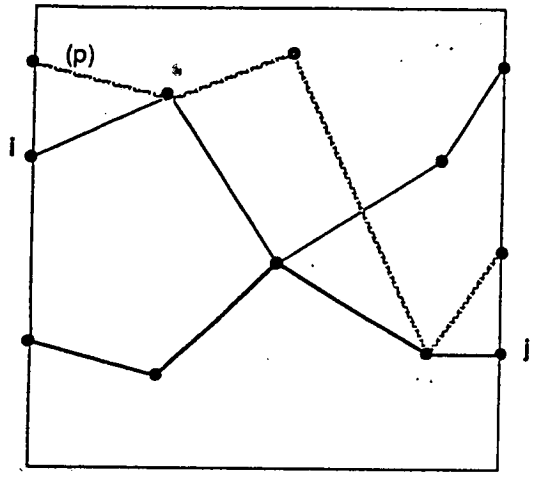

Fig 3 The Banyan Property

If path (p) exists, there are two different paths from i to j.

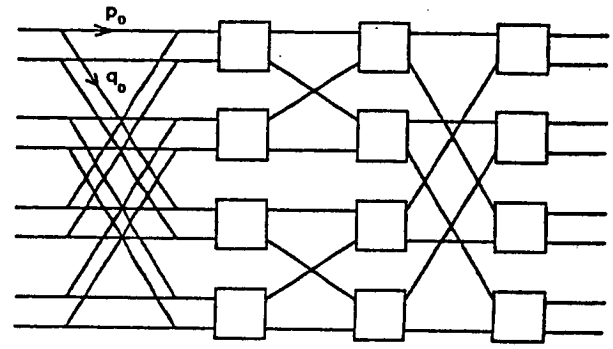

Fig 4a 1-Extention of the Indirect Binary Cube

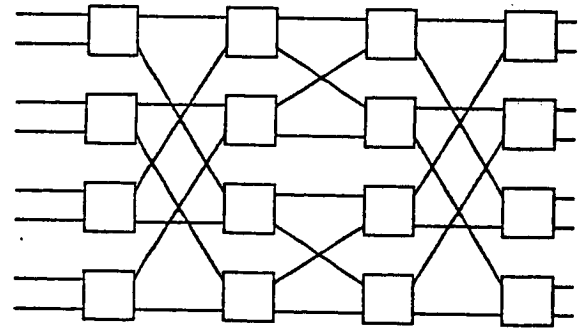

Fig 4b Representation with Switching Cells

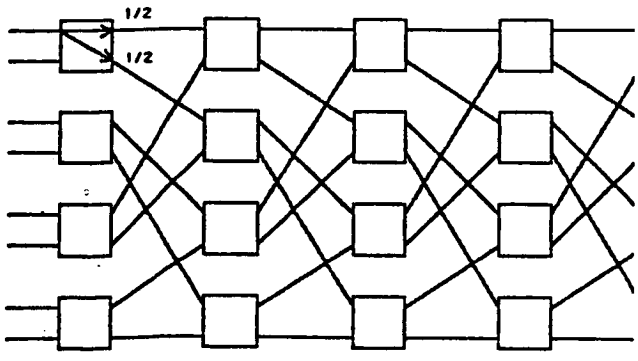

Fig 5 1-Extended OMEGA Network

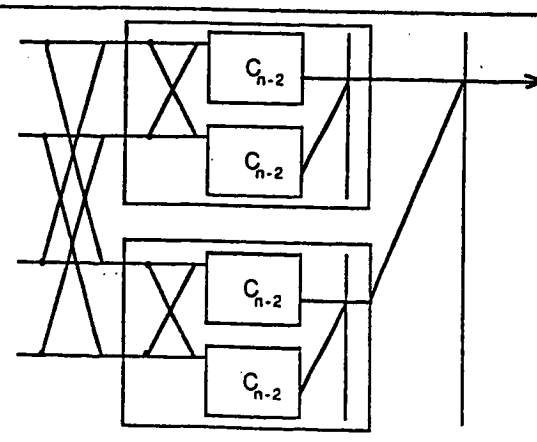

Fig 6 Recursive Construction of the Extentions

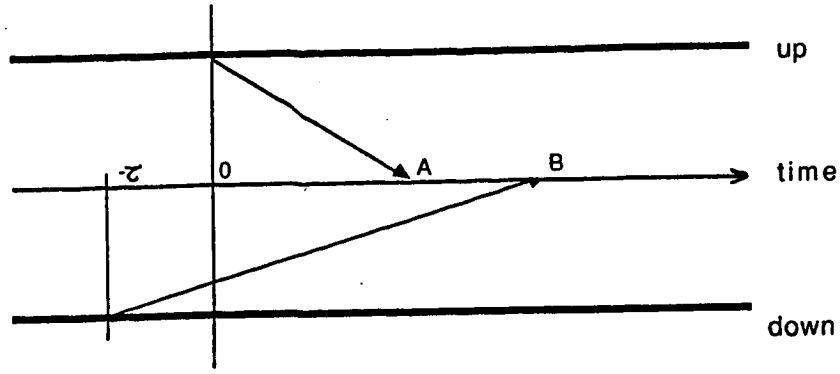

Fig 7 Resequencing time of a customer

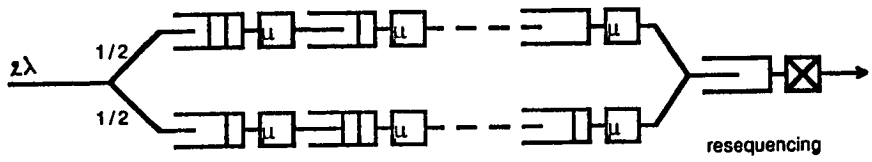

Fig 8

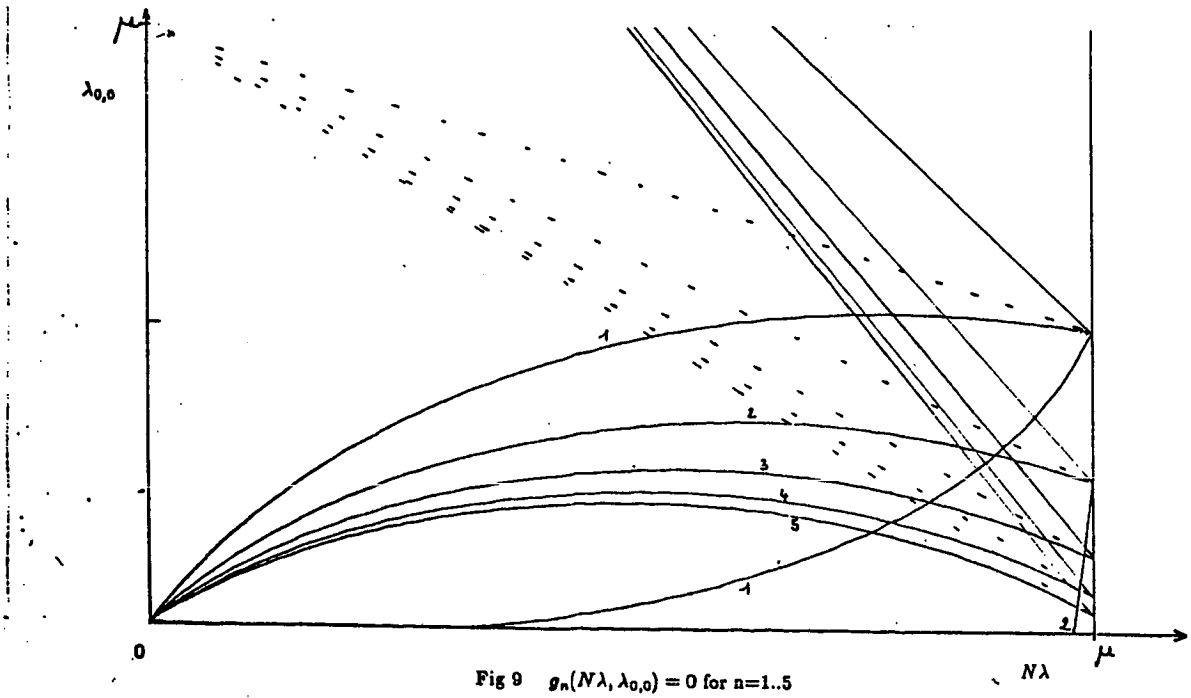

Fig 9 $g_n(N\lambda, \lambda_{0,0}) = 0$ for n=1..5

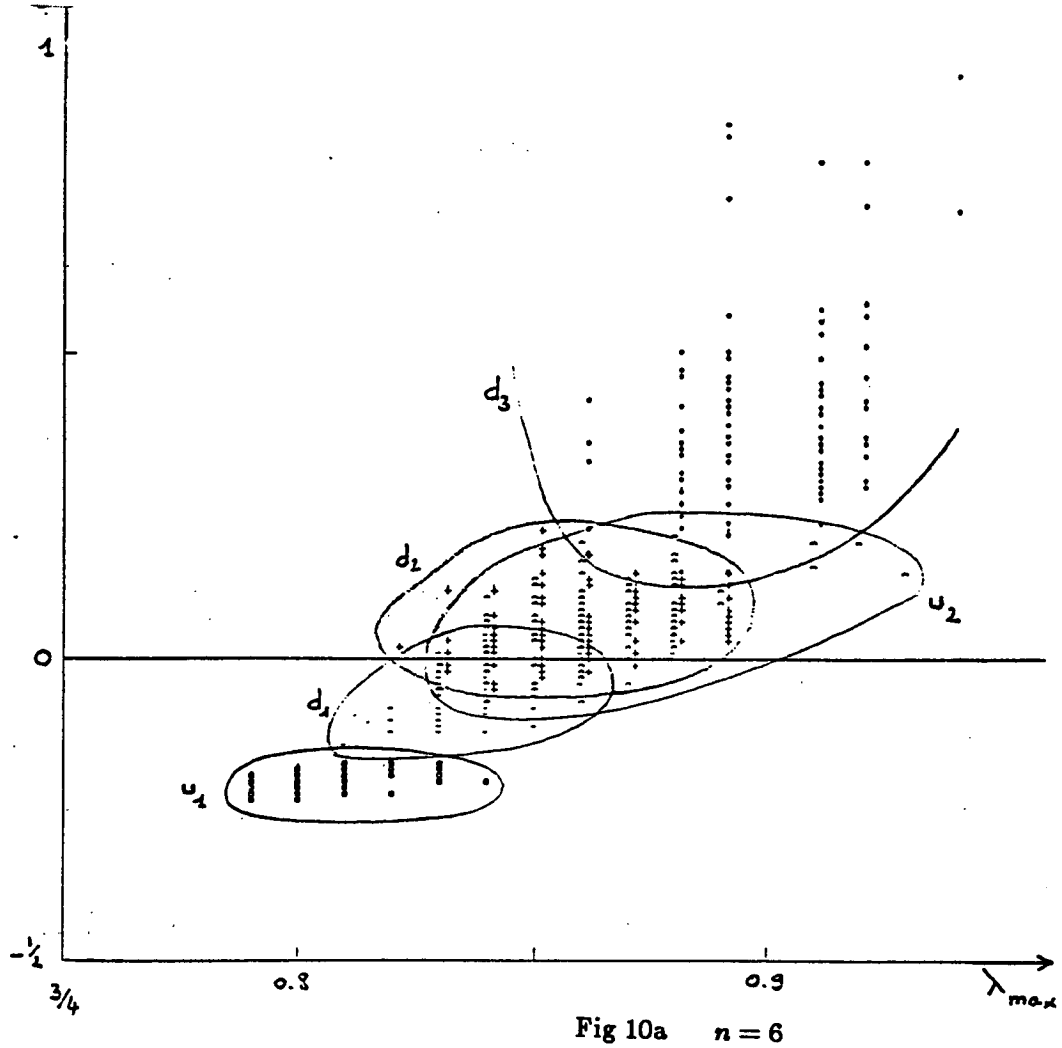

Fig 10a $n = 6$

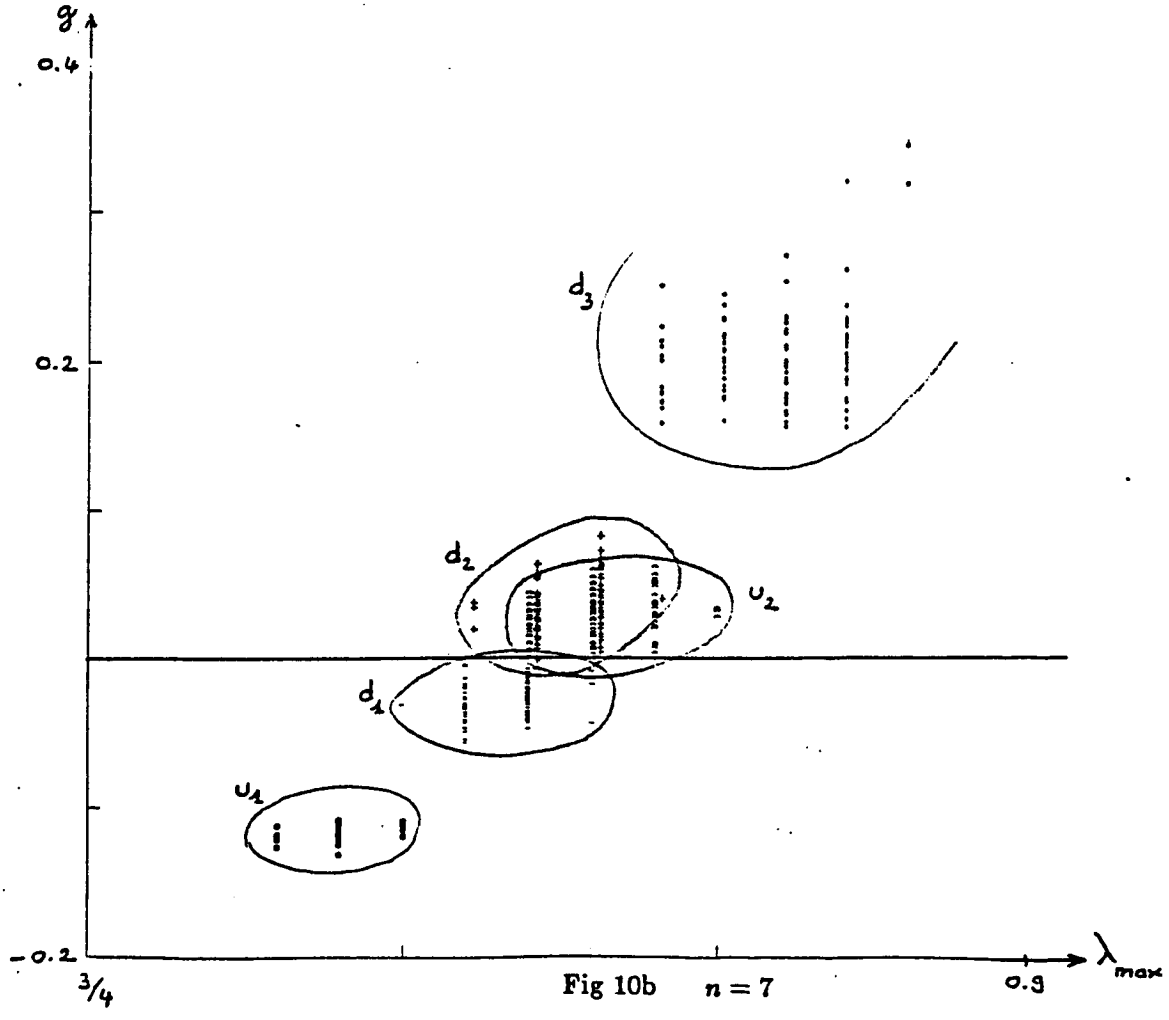

Fig 10b $n = 7$

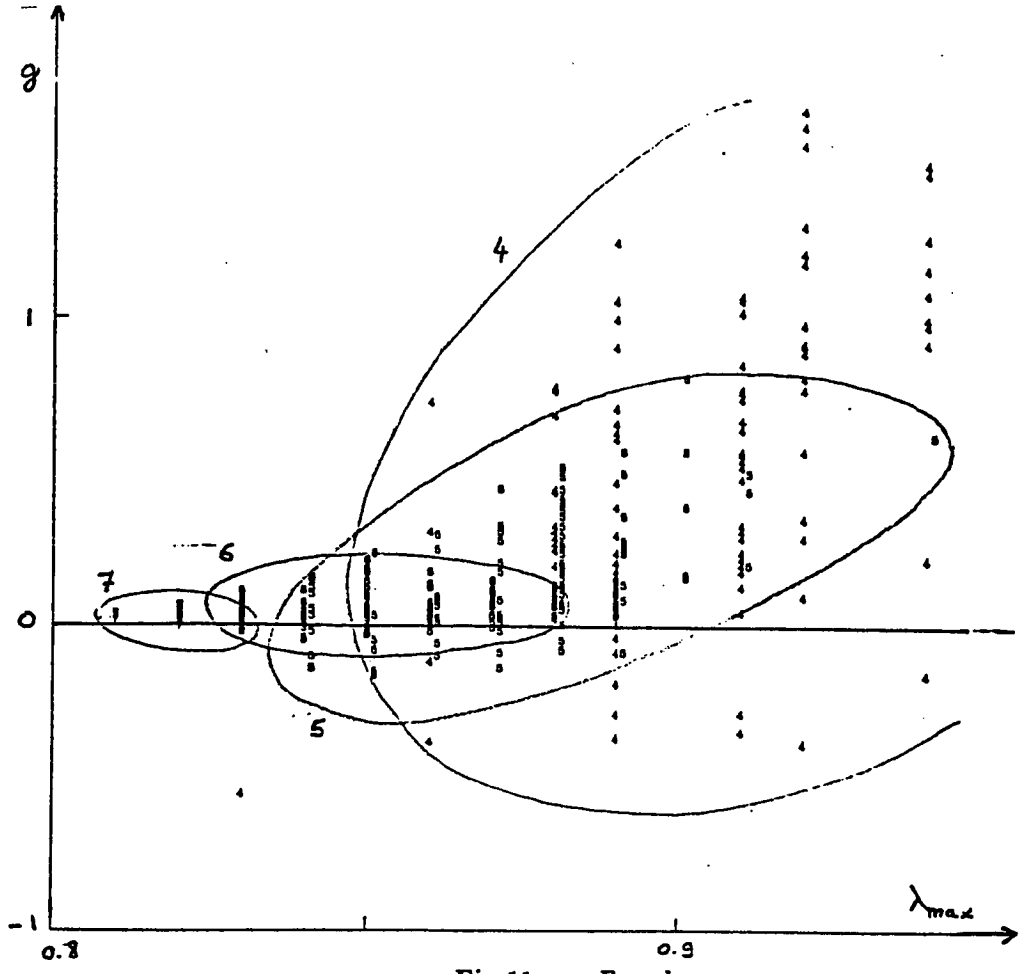

Fig 11a $F = d_2$

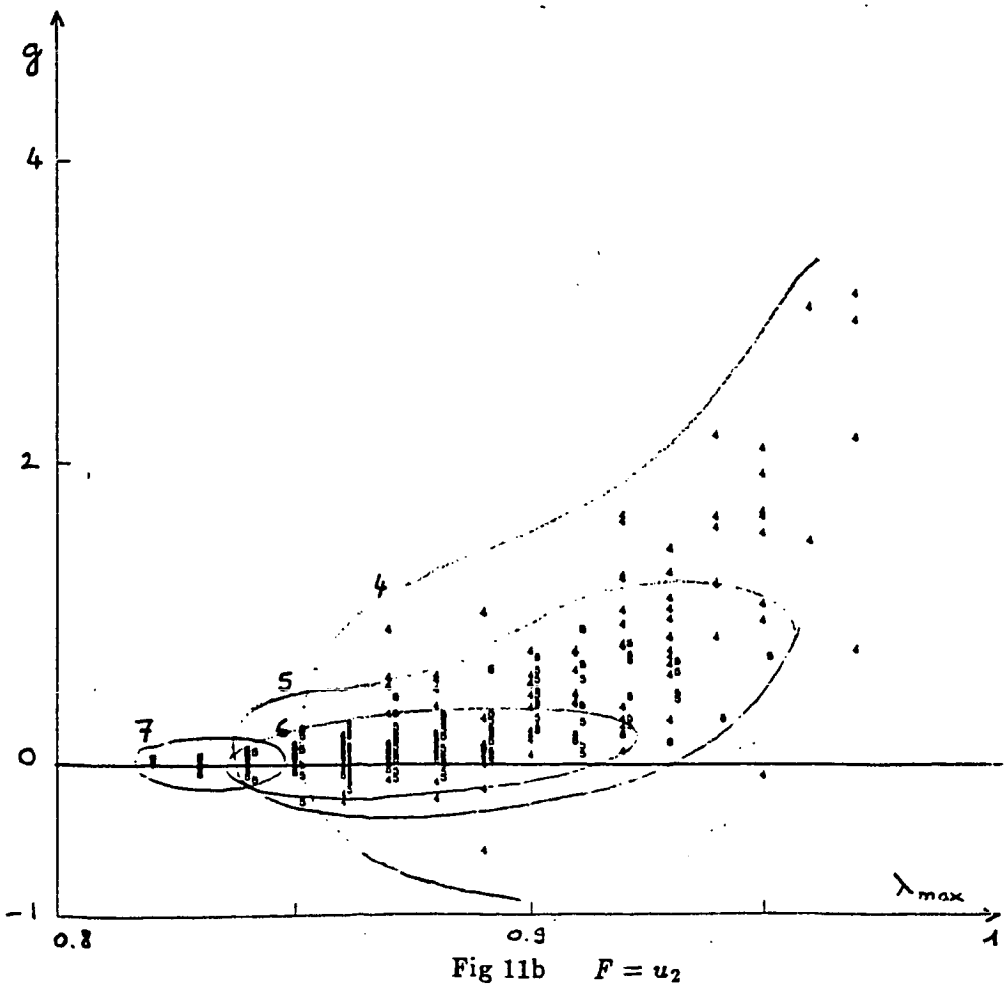

Fig 11b $F = u_2$

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique